ELSEVIER International Journal of Approximate Reasoning 22 (1999) 93–107

INTERNATIONAL JOURNAL OF APPROXIMATE REASONING

Improving fuzzy systems identification with data transformations

Armin Shmilovici [a,*], Joseph Aguilar-Martin [b]

[a] *Deptartment of Industrial Engineering, Ben-Gurion University, P.O. Box 653, 84105 Beer Sheva, Israel*

[b] *LEA-SICA, LAAS CNRS, 7 avenue du colonel Roche, 31077 Toulouse, France*

Received 1 October 1998; accepted 1 February 1999

Abstract

A practical problem in the identification of fuzzy systems from data, is the design and the tuning of the membership functions. We demonstrate that if the data is properly transformed before the identification process, the resulting fuzzy model can be improved to the point it may not need a further tuning. The significance of the data transform can be validated using statistical methods. The method is demonstrated on a time series prediction problem, using the Box–Cox transform. © 1999 Elsevier Science Inc. All rights reserved.

Keywords: Fuzzy systems identification; Data transformations; Applied statistics; Variance heterogeneity; Box–Cox transform

1. Introduction

The main advantage of fuzzy models, is their ability to describe expert knowledge in a descriptive, human like way, in the form of simple rules using linguistic variables. The theory of fuzzy sets [1,2] allows the existence of uncertainty due to vagueness (or fuzziness) rather than due to randomness. When using fuzzy sets, accuracy is traded for complexity – fuzzy logic models do not

* Corresponding author. Tel.: +972-7-646-1434; fax: +972-7-647-2958; e-mail: armin@bgumail.bgu.ac.il

PII: S0888-613X(99)00013-4

need an accurate definition of the system (in terms of the parameters). This results in a natural reduction in the number of variables and states that describe the system.

In the practice of building fuzzy models, there are two complementary and non-exclusive methods to construct a decision or a control surface by means of fuzzy inference, where fuzzy membership functions are used as the building blocks. The first is based on representing the expertise of skilled human operators. The second makes use of the universal approximation property [3] of fuzzy sets to build a model from data gathered from the actual system. An actual understanding of the physical phenomena is not a prerequisite, thus it makes fuzzy models attractive as a general-purpose non-linear model building procedure. This paper will investigate the second method from a fuzzy regression like model perspective.

When constructing a fuzzy model based on a data set derived from a real experiment, a perfect fit between model and data never exists. Thus, there is a need for determining the quality of the model, and its relevancy to the application. If the quality is not sufficient, the model is subjected to a further computationally expensive iterative optimization process [4,5], until their performance is considered satisfactory.

The novelty and contribution of this paper is in proposing a method to improve the quality of fuzzy models before performing a complex optimization process. The main idea is to transform the data to a different range of values before performing the identification process. Here, the statistical aspects of the transform are considered, rather than the function approximation aspects analyzed in Ref. [11]. Since fuzzy models are user oriented, the specific transform used will depend on the type of the application. The identification process will be performed on the transformed data, for a standard family of membership functions (the splines) which have a good accuracy-complexity tradeoff [6]. An inverse transformation will be used to test the final quality of the resulting fuzzy model. A dual description of the problem, is to design a transform for a *standard* set of membership function, such that the fitting error between model and data is minimized.

The outline of rest of the paper is as follows: Section 2 introduces the rational behind data transforms from a statistical point of view, Section 3 introduces the fuzzy models used, Section 4 presents an application example, and Section 5 concludes with some discussion.

2. Fuzzy models and statistical transformations

Fuzziness stems from either the complexity of the process itself, or from the perceptions of human beings. Therefore the fuzzy model is strongly user-oriented and its construction reflects this phenomenon. The identification problem for fuzzy models can be conveniently viewed in the general setting of systems identification as already formulated in the literature [7]. In common use it is treated as comprising a sequence of steps such as structure identification, parameter estimation and model validation. The common procedure to build a fuzzy model based on examples includes the following sequence:

Step 1: Partition the input and output spaces of the given numerical data into fuzzy regions.

Step 2: Generate fuzzy rules from the given data.

Step 3: Assign a degree of confidence for each one of the generated rules for resolving conflicts.

Step 4: Create a combined fuzzy rule base using knowledge from human experts.

Step 5: Determine the mapping from input space to output space (the inference mechanism)

Step 6: Optimize and improve the model.

In practical applications, Step 6 is usually the most computationally expensive, since it requires a simultaneous non-linear optimization of many coupled parameters.

In the general setting of non-linear memoryless relations expressed by means of fuzzy models, the function that represents the data can be formulated as

$$y_i = f(\bar{x}_i, \bar{\theta}) + \varepsilon_i, \quad i = 1, 2, \ldots, n, \tag{2.1}$$

where $f(\bar{x}_i, \bar{\theta})$ is the non-linear function (surface) defined by the fuzzy model of the relation between input and output data pairs $\{\bar{x}_i, y_i\}$, ε_i is the residual modeling error, and $\bar{\theta}$ is the parameter vector of the fuzzy model, which is found in the identification process. In the setting of the fuzzy models of Section 3, the parameter vector determines the shapes of the fuzzy sets, and the conclusion parts of the fuzzy rules.

From a statistical point of view, when constructing a model based on a data set coming from any real experiment, a perfect fit between model and data never seems to exist. In the statistical community, the sum of squared distances is commonly used to measure the fit between a model and the data upon it is built. In order to accept/reject a model, the mean squared distance is compared to a critical ratio and the model is rejected if its value is above the critical ratio (F test).

Some of the underlying assumptions in the statistical literature [8] are:

(a) The parameters $\bar{\theta}$ of Eq. (2.1) are determined by minimization of $\sum_i [f(\bar{x}_i, \bar{\theta}) - y_i]^2$.

(b) Approximate normality of the residuals ε_i.

(c) Homogeneity of the distribution of ε_i.

Neither of the above assumptions can be substantiated for many of the fuzzy models presented in the literature:

(a) The model may result from expert knowledge, or from a process of cost functional minimization (as is common in control system design).
(b) The approximate normality of ε_i cannot be guaranteed. Considering that fuzzy systems are function approximators, in certain cases (e.g., low noise data) most of the errors may result from the approximation error, which will not be normally distributed. (in statistical language, the model's bias maybe larger than the variance)
(c) Fuzzy systems are composed of different rules which span different ranges of the input-output mapping, and may result from different sources (e.g. experts) with different accuracy. Thus, the distribution is not uniform for all the intervals of the input data, and may depend on the domain of the fuzzy rules activated (e.g., it is a common design feature in fuzzy controllers that the error near the controlled set-point will be smaller than the error elsewhere).

Assumptions b, c boil down to the concept of *variance heterogeneity.* Variance heterogeneity means that we cannot assume in the optimization process, that the residuals ε_i are independent and identically distributed (i.i.d). Thus, some of the main justifications for using least square optimization methods are not valid. Unfortunately, most fuzzy systems identification and optimization methods, which perform a global minimization of the sum of square residuals, ignore this problem.

In the statistical literature [8], two possible cures are suggested for handling variance heterogeneity:

1. The "power transform both sizes" model, in which we are looking for a transformation function h with a parameter λ

$$h(y_i, \lambda) = h[f(\bar{x}_i, \bar{\theta}), \lambda] + \varepsilon_i \tag{2.2}$$

such that the residuals ε_i are i.i.d.

2. The "Power Transformed Weighted Least Square" model, in which a specific model is assumed for the distribution of the residuals, i.e. for the heterogeneity of their variance.

Iterative procedures are used to determine which transformation function (or weighting of the residuals) will produce almost i.i.d. residuals.

In the statistical literature, method 2 has often been preferred to method 1 when the model (2.1) is linear. In the framework of fuzzy systems, when dealing with membership functions with the partition of one property (i.e. the sum of the fuzzy set at each point in the domain is equal to 1), method 1 is preferred, since it introduces a change in the shape of the fuzzy set, which can have direct intuitive interpretation.

In this paper, we effectively propose that using Eq. (2.2) and actively solving for λ (i.e. one non-linear optimization) can result in significant improvement of the fuzzy model (2.1), to the point that the multidimensional non-linear optimization of $\bar{\theta}$ can sometimes be eliminated.

Example: Optimization of a fuzzy controller:

In a regulation control, where the response of the plant is to be kept near a given setpoint, the accuracy near the setpoint is more important to the overall quality of the controller than the accuracy near the edges of the domain. Thus, the fuzzy sets should be denser near the setpoint than near the edges. Suppose that instead of choosing an arbitrary fuzzy set and optimizing it for improved accuracy, we will use the scaled hyperbolic tangent as the transform function that will improve the accuracy of the controller near the setpoint.

$$h(x, \lambda) = \mathrm{tangh}(\lambda x) = \frac{e^{\lambda x} - e^{-\lambda x}}{e^{\lambda x} + e^{-\lambda x}}, \quad \lambda > 0. \tag{2.3}$$

Fig. 1 Presents a single input single output system for the case where the setpoint is 0 and the data is scaled to the range [−1, 1]. The scaled hyperbolic tangent function with $\lambda = 3$ provides increased resolution around 0. If we identify a *standardized fuzzy set* over the domain of transformed data (e.g. the quadratic spline fuzzy set of the appendix) – as in Fig. 1(bottom); then our fuzzy system will effectively use the fuzzy set of Fig. 1(top) (found by applying the inverse scaled hyperbolic tangent on the standardized fuzzy sets). The improved accuracy obtained near the setpoint when optimizing λ with respect to the variance heterogeneity will lead to a consistent improvement in the determination of the function.

There are several methods to optimize for the best the parameter λ for a given dataset, such as the minimization of the variance in Eq. (2.2), and maximum likelihood estimation [8]. While many other transformation functions can be proposed, most of them cannot eliminate completely the problem

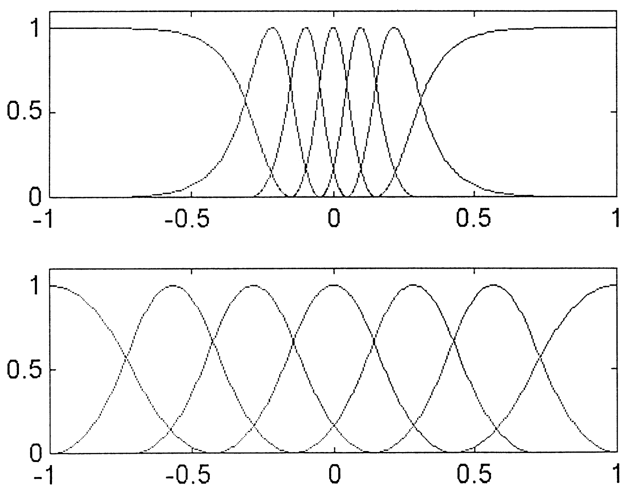

Fig. 1. Top: original data; Bottom: transformation with tangh($3x$).

of variance heterogeneity and be simple to compute for a general dataset. Section 4 presents the use of the Box–Cox transform, which is known to eliminate variance heterogeneity [8].

3. Regression-like fuzzy models

The fuzzy system considered in this paper is comprised of four basic elements: fuzzifier, fuzzy rule base, fuzzy inference engine, and defuzzifier. We consider multi-input single-output fuzzy systems: $f : U \subset \boldsymbol{R}^n \to V \subset \boldsymbol{R}$, where $U = U_1 \times U_2 \times \cdots \times U_n \subset \boldsymbol{R}^n$ is the input space and $\mathrm{V} \subset \mathbf{R}$ is the output space. A multi-output system can be represented as a group of single-output systems.

A rule, is a proposition that implies another proposition. In this paper, the *fuzzy rule base* consists of a set of linguistic rules in the form of "IF a set of conditions are satisfied THEN a set of consequences are inferred". Assume that there are N rules of the following form:

$$\begin{aligned} &R_i : \text{IF } x_1 \text{ is } A_{i1} \text{ and } x_2 \text{ is } A_{i2} \text{ and} \ldots \text{and } x_n \text{ is } A_{in} \text{ THEN } y \text{ is } C_i, \\ &\quad i = 1, 2, \ldots, N, \end{aligned} \tag{3.1}$$

where x_j $(j = 1, 2, \ldots, n)$ are the input variables to the fuzzy system, y is the output variable of the fuzzy system, and the fuzzy sets A_{ij} in U_j and C_j are linguistic terms characterized by fuzzy membership functions $A_{ij}(x_j)$ and $C_i(y)$, respectively. Each rule R_i can be viewed as a *fuzzy implication* (relation) $A_i = A_{i1}xA_{i2}x, \ldots, xA_{in} \to C_i$, which is a fuzzy set in $U \times V = U_1 \times U_2 \times \cdots \times U_n \times V$ with membership function $R_i(X_i, y) = A_{i1}(x_1)^* A_{i2}(x_2)^* \cdots {}^* A_{in}(x_n)^* C_i(y)$, and* is the T norm [1], $\bar{x} = (x_1, x_2, \ldots, x_n) \in U$ and $y \in V$.

The fuzzy models used in this paper are restricted to singleton fuzzifier, sup-product inference engine, and centroid defuzzifier. In Ref. [3] it is proved that in that case, the fuzzy system can be represented as a linear combination of Fuzzy Basis Functions (FBF)

$$f(\bar{x}) = \sum_{i=1}^{N} g_i(\bar{x}) y_i, \tag{3.2}$$

where y_i $(i = 1, 2, \ldots, N)$ is the point for which $C_i(y)$ achieves its maximum value (i.e. $C_i(y_i) = 1$ under the assumption that C_i is a normalized fuzzy set).

For each rule in the fuzzy rule base there is a corresponding FBF

$$\frac{A_i(x)}{\sum_{i=1}^{N} A_i(x)} = \frac{\prod_{j=1}^{n} A_{ij}(x_j)}{\sum_{i=1}^{N} \left[\prod_{j=1}^{n} A_{ij}(x_j) \right]}, \tag{3.3}$$

where $A_i(\bar{x}) = A_i(x_1, x_2, \ldots, x_n) = \prod_{j=1}^{n} Aij(xj), (i = 1, 2, \ldots, N)$.

In the identification problem, it is assumed that there are $k = 1, \ldots, T$ input-output pairs $(\bar{x}(k), f(\bar{x}(k)))$. Our task is to design a FBF expansion $\hat{f}(\bar{x})$ such that the error between $f(\bar{x})$ *and* $\hat{f}(\bar{x})$ is minimized. We can arrange Eq. (3.2) in the following regression-like form

$$\bar{f} = G \cdot \bar{a} + \bar{e}, \tag{3.4}$$

where

$$\bar{f} = [f(\bar{x}(1), \ldots, f(\bar{x}(T))], \quad G = [\bar{g}_1, \ldots, \bar{g}_N],$$

where

$$\bar{g}_i = [g_i(x(1)), \ldots, g_i(x(T))]^{\mathrm{T}},$$

and $\bar{a} = [a_1, \ldots, a_N]^{\mathrm{T}}$ a vector of unknown regressors, and $\bar{e} = [e_1, \ldots, e_T]^{\mathrm{T}}$ is the identification error. The direct Least-Squares (regression) solution of (3.4) may not be possible in the common case where $rank(G) < n$ (i.e. there is linear dependence between membership functions). In that case we have to use other methods, such as the orthogonal matching pursuit of Ref. [10].

While the transformation methods in this paper will benefit any fuzzy sets A_{ij}, we propose the use of the quadratic spline basis functions described in the appendix, due to their desirable smoothness and computational properties. Multidimensional basis functions can be built with the tensor product of the single dimension basis function.

4. Time series prediction using the Box–Cox transform

The Box–Cox transformation (discussed in Ref. [8]) is used for handling variance heterogeneity, using the method of Eq. (2.2).

$$h(y, \lambda) \equiv y^{(\lambda)} = \begin{cases} \dfrac{y^{\lambda} - 1}{\lambda}, & \lambda \neq 0, \\ \log_e y, & \lambda = 0, \end{cases} \quad \text{valid for use when } y > 0. \tag{4.1}$$

Since λ itself is unknown, it has to be estimated as $\hat{\lambda}$, the value that minimizes the mean square error of the fit between the transformed data and its fuzzy model. In applications, to simplify the optimization procedure, rather then finding $\hat{\lambda}$ itself, it is possible to take advantage of the existence of a small number of values of λ that cover sufficiently most of the situations, and then to optimize among that reduced discrete set. The set of "preferred" values for the Box–Cox transformation is taken as $\{-1, -1/2, 0, 1/2, 1\}$.

Following Ref. [8], the concentrated log-likelihood function Eq. (4.2) can be used as the criterion to be minimized for an unbiased parameter estimation of λ. Since we are only interested in one of the above five possible values of $\hat{\lambda}$, a complex non-linear minimization process can be avoided:

$$M(\lambda) = -\frac{n}{2} \log \hat{\sigma}_{\lambda}^2 + (\lambda - 1) \sum_{i=1}^{n} \log y_i \tag{4.2}$$

where n is the sample size and $\hat{\sigma}_\lambda^2$ is the variance of the residual error of the (fuzzy) model for the transformed data:

$$\hat{\sigma}_\lambda^2 = \frac{1}{n}\sum_{i=1}^{n}\left[y^{(\lambda)} - f(x_i, \bar{\theta}_\lambda)^{(\lambda)}\right]^2, \tag{4.3}$$

where $f(x_i, \bar{\theta}_\lambda)^{(\lambda)}$ is the fuzzy model linking transformed measured inputs to transformed measured outputs, and $\bar{\theta}_\lambda$ are the parameters identified for that fuzzy model, given a specific λ. An approximate $100(1-\alpha)\%$ confidence interval for λ is given by

$$\left\{\lambda : M(\lambda) \geqslant c_\alpha = M(\hat{\lambda}) - \frac{1}{2}\chi_1^2(\alpha)\right\}, \tag{4.4}$$

where $\chi_1^2(\alpha)$ is the Chi-square statistic with 1 degree of freedom. $\chi_1^2(0.1) = 0.0158$.

Example: The Mackey–Glass time-delay differential equation is used as a benchmark problem in the neural network and fuzzy modeling communities [3,4]

$$\dot{x}(t) = \frac{0.2x(t-\tau)}{1 + x^{10}(t-\tau)} - 0.1x(t). \tag{4.5}$$

Its solution produces a chaotic times series for a sufficiently large time delay τ. To obtain a solution at the integer points, we applied the Matlab command ode23 for the second order Runge–Kutta differential equation solver. The equation was integrated from one integration point to the next, with an accuracy of 10^{-3}, with initial conditions $x(0) = 1.2$, $\tau = 17$, and $x(t) = 0$ for $t < 0$. Fig. 2 shows the time series for $t \in (100, 1100)$ s.

Following the experiment in Ref. [4], the values of $\bar{x} = x(t-18)$, $x(t-12), x(t-6), x(t)$ were used to predict the future value of $f = x(t+6)$. 1000 4-input-1-output data pairs $\{\bar{x}(k), f(k)\}$ were extracted from the time series, starting from $t = 100$. The first 500 data pairs were used for training the fuzzy system, while the last 500 data pairs where used for validating the quality of the fuzzy model.

Similar to the system of Ref. [4], the fuzzy system here was designed to have two membership functions, of the edge quadratic spline defined in the appendix, for each input in the domain $x(t) \in [0.2, 1.6]$ (Fig. 3). The training data was arranged as in Eq. (3.4), and the Orthogonal Matching Pursuit algorithm of Ref. [10] was used to identify $2^4 = 16$ fuzzy rules.

Comparing the predicted time series with the original time series (Fig. 4), the fuzzy model is quite good without any further optimization, and there is no apparent difference in the error (Fig. 5) between the training set and the testing set.

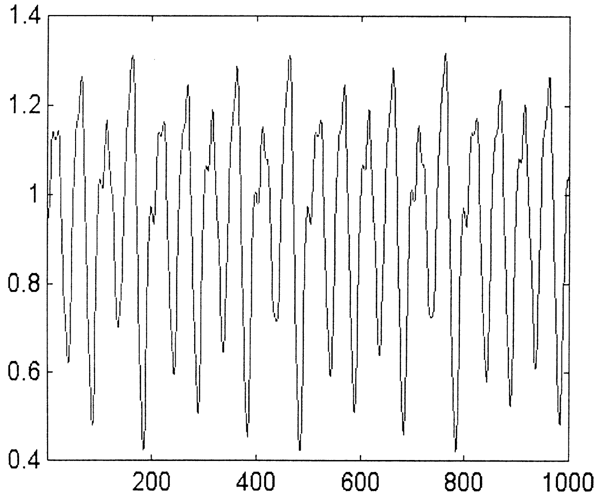

Fig. 2. 1000 unit samples of the Mackey-Glass chaotic time series.

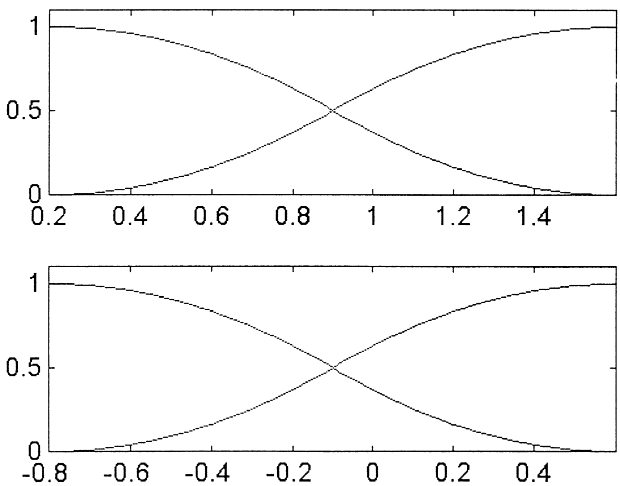

Fig. 3. Top: original; Bottom: linear transformation $\lambda = 1$.

The training set was transformed with five λ values of the Box–Cox transform, with 2 quadratic spline fuzzy sets designed for each transformed input. Table 1 present the results of testing the five models, and Figs. 3, 6, 7, 8, 9 present the fuzzy sets used, and their inverse transform.

As we can see from Table 1, and according to Eq. (4.4), the value $\lambda = 0$, (i.e. log transformation), turned out to be significantly larger than the next value

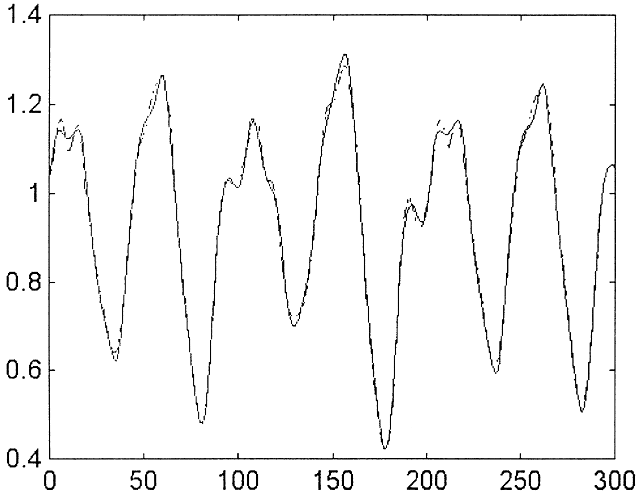

Fig. 4. First 300 samples of original series (solid) and predicted series (dashed).

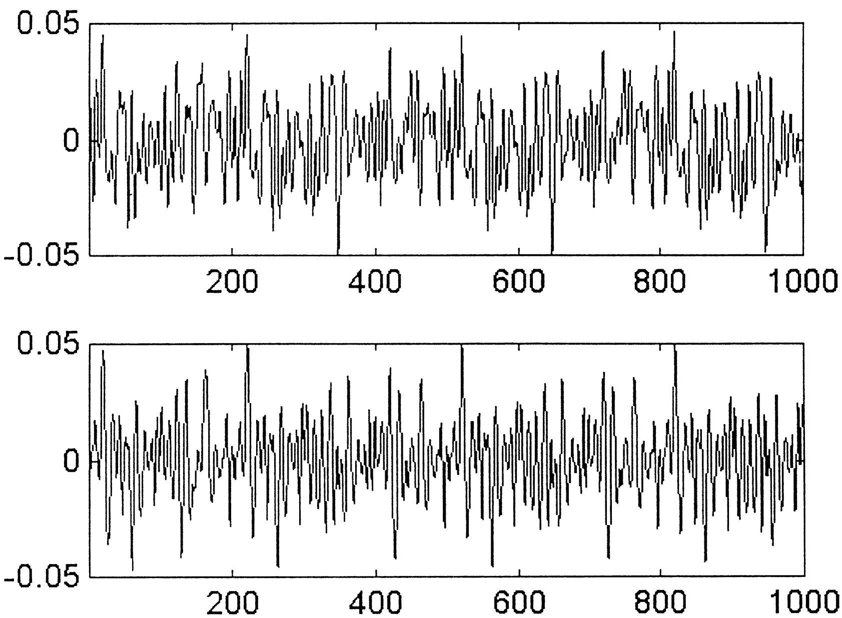

Fig. 5. Prediction errors for learning set (1:500) and testing set (501:1000); Top: prediction error without data transform; Bottom: prediction error for log transform $\lambda = 0$.

$\lambda = 1$, (i.e. linear scaling of the data). Since the domains of the fuzzy sets were designed rather arbitrarily, we cannot really conclude which transformation is best. Looking at the prediction graphs (the dashed lines in Fig. 10) and the

Table 1
Results of the numerical experiments

λ	Range	$\hat{\sigma}_\lambda$	$M(\lambda)$	Fig. No.
1	(−0.8,0.6)	0.0170	2.036×10^3	3
1/2	(−1.0,0.5)	0.0239	1.894×10^3	9
0	(−1.1,0.5)	0.0181	2.059×10^3	6
−1/2	(−1.4,0.5)	0.0505	1.574×10^3	8
−1	(−1.7,0.5)	0.0806	1.366×10^3	7

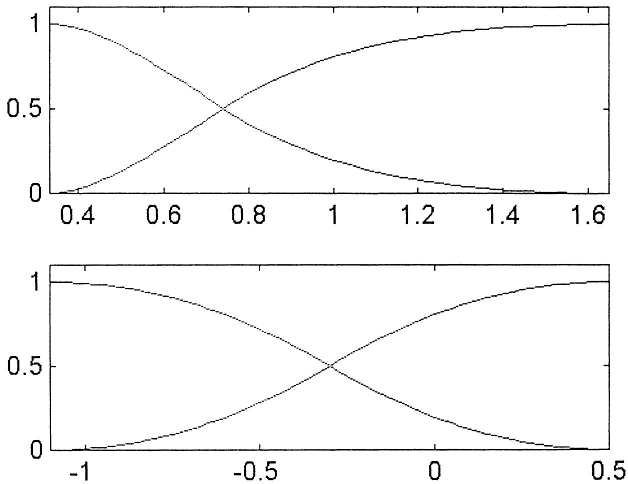

Fig. 6. Top: original axis; Bottom: log transformed axis $\lambda = 0$.

prediction errors (Fig. 5), the difference is very small if any. While in both cases, further tuning of the fuzzy sets is possible, the potential improvement is indicated to be small.

In total, 16 linear parameters and 1 non-linear parameter were used for training the fuzzy system. For comparison, in Ref. [4], 80 linear and 24 non-linear parameters were used. The tuning process of the 104 parameters was indeed able to further reduce the prediction error; yet, a computationally expensive procedure was needed. Our method succeeded in finding membership functions, which may not have the optimum shape, yet provide a sub-optimal performance at a significant reduction in the computational cost. This kind of trade-off may be desired in many types of engineering applications.

Note that the Box–Cox transform is defined only for $y > 0$. The data has to be translated, or other transformations have to be used in case this condition is not met.

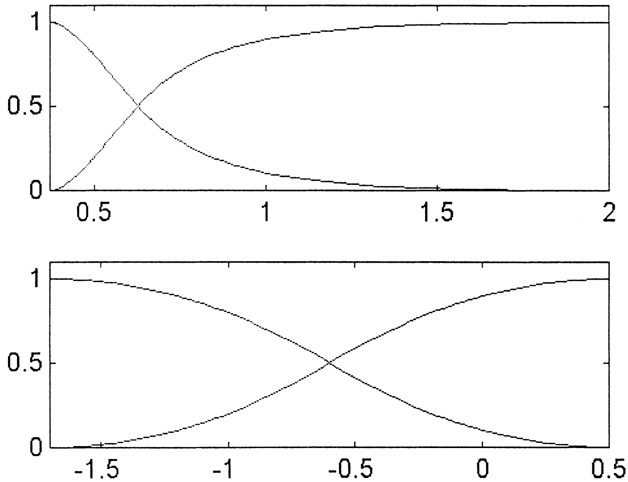

Fig. 7. Top: original axis; Bottom: transform with $\lambda = -1$.

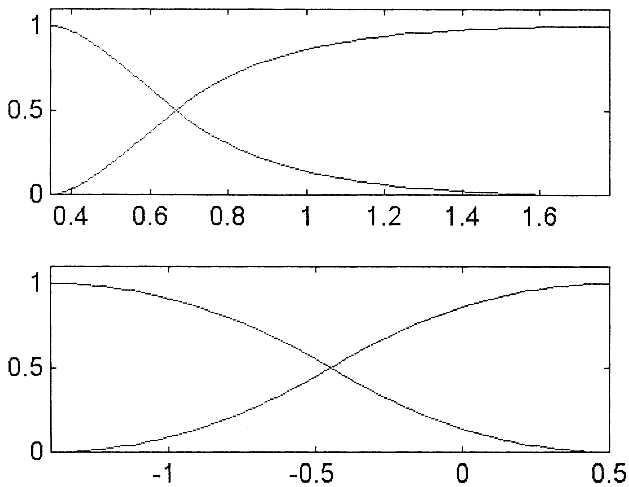

Fig. 8. Top: original axis; Bottom: transform with $\lambda = -1/2$.

5. Conclusions and discussion

Many of the fuzzy models used today ignore the issue of variance heterogeneity in their tuning process. A data transformation method – the Box–Cox transform – adapted from the statistical literature, was proposed for the design and tuning of fuzzy sets. This transform solves the problem of variance heterogeneity and improves the quality of the fuzzy model at a modest computational

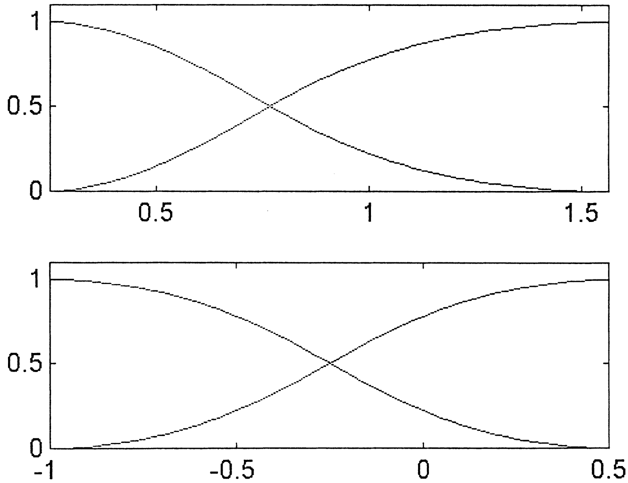

Fig. 9. Top: original axis; Bottom: transform with $\lambda = 1/2$.

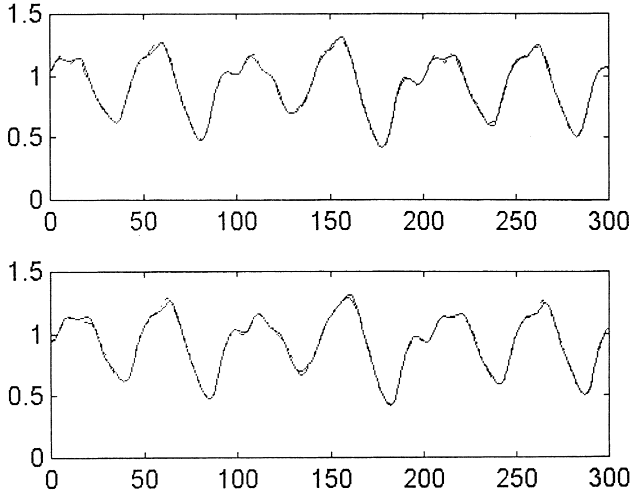

Fig. 10. Top: original series and predicted series (dashed); Bottom: original series and predicted series after transformation and inverse transformation (dashed).

cost. The benefits of the transform were demonstrated on a time series prediction problem.

The notion of transforming the data to improve the accuracy of the fuzzy sets at specific domains of interests is mathematically sound and intuitively appealing. Unfortunately, the currently available statistical transformations, were developed especially for linear regression models, thus may not be appropriate for many applications of fuzzy models. For example, in control applications, improved accuracy is needed near the controller setpoint. While transforming the data with the hyperbolic tangent function can improve the accuracy near the setpoint, no currently available statistical test can give a confidence interval for the transformation parameters. Thus, further work is needed to develop new transformation methods.

Appendix A. The spline basis functions

The B-spline of order n is denoted by $\beta^n(x)$. It is a piecewise polynomial of degree n. B-splines are symmetric bell-shaped functions; they have a simple analytical form [9].

The quadratic spline (QS) $\beta^2(x)$ is composed of three parabolic sections:

$$\beta^2(x) = \begin{cases} \frac{1}{2}x^2, & \text{if } 0 \leqslant x \leqslant 1, \\ -x^2 + 3x - \frac{3}{2}, & \text{if } 1 \leqslant x \leqslant 2, \\ \frac{1}{2}(x-3)^2, & \text{if } 2 \leqslant x \leqslant 3, \\ 0, & \text{otherwise.} \end{cases} \tag{A.1}$$

Special splines are designed for the boundaries of the domain:

$$\beta_L^2(x) = \begin{cases} 1 - \frac{1}{2}x^2, & 0 \leqslant x \leqslant 1, \\ \frac{1}{2}(x-2)^2, & 1 \leqslant x \leqslant 2 \text{ left side}, \\ 0, & \text{otherwise.} \end{cases} \tag{A.2}$$

$$\beta_R^2(x) = \begin{cases} \frac{1}{2}x^2, & 0 \leqslant x \leqslant 1, \\ -\frac{1}{2}(x)^2 + 2x - 1, & 1 \leqslant x \leqslant 2 \text{ right}, \\ 0, & \text{otherwise.} \end{cases} \tag{A.3}$$

Theorem. *A fuzzy system built with spline fuzzy sets can universally approximate any function $f \in L^2(R)$ with a bounded error term.*

$$f(x) = \sum_i a_i \beta^2(x - i) + E \tag{A.4}$$

where $\beta^2(x)$ is the spline of order 2 and E is a bounded error term

$$||E||_\infty \leqslant \frac{0.0481}{3!} h^3 ||f^{(3)}(x)||_\infty \tag{A.5}$$

where h is the inverse of the number of basis functions per unit length. The proof is in Ref. [6]. *Basically, the FBF of* Eq. (A.1) *are proved to be a normalized version of the spline basis function expansion of a smooth function $f \in L^2(R)$. Spline basis functions of a given smoothness space are known to have the least approximation error for that smoothness space* [9]. *A quadratic spline, which can preserve the first derivative, is sufficient for most applications.*

References

[1] H.J. Zimmermann, Fuzzy Set Theory, Kluwer Academic Publishers, Dordrecht, 1991.
[2] G.J. Klir, T.A. Folger, Fuzzy Sets, Uncerainty and Information, Prentice-Hall, Englewood Cliffs, NJ, 1988.
[3] L.X. Wang, Adaptive Fuzzy Systems and Control-Design and Stability Analysis, Prentice Hall, Englewood Cliff, NJ, 1994.
[4] J.S.R. Jang, ANFIS: Adaptive network based fuzzy inference system, IEEE Trans. On Systems Man and Cybernetics 23 (3) (1993) 665–685.
[5] M. Brown, C. Harris, Neurofuzzy Adaptive Modeling and Control, Prentice-Hall, Englewood Cliffs, NJ, 1994.
[6] A. Shmilovici, O.Z. Maimon, On the solution of differential equations with fuzzy spline wavelets, Fuzzy Sets and Systems. 96 (1) (1998) 77–99.
[7] W. Pedrycz, Fuzzy Control and Fuzzy Systems, Wiley, New York, 1993.
[8] G.A.F. Seber, C.J. Wild, Non-Linear Regression, Wiley, New York, 1989.
[9] L.L. Schumaker, Spline Functions Basic Theory, Wiley, New York, 1981.
[10] A. Shmilovici, O. Maimon, Fuzzy systems identification with orthogonal matching pursuit, In: Proceedings of the Fifth IEEE International Conference on Fuzzy Systems (FUZZ-IEEE'96), New-Orleans September 8–11 1996 pp. 2059–2064.
[11] Z.H. Mao, Y.D. Li, X.F. Zhang, Approximation capability of fuzzy systems using translations and dilations of one fixed function as membership functions, IEEE Transactions on Fuzzy Systems 5 (1997) 468–473.